REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
July 1996

3. REPORT TYPE AND DATES COVERED
Technical Report

4. TITLE AND SUBTITLE
THE PESKI INTELLIGENT USER INTERFACE

5. FUNDING NUMBERS

6. AUTHOR(S)
Robert A. Harrington, Sheila Banks, and Eugene Santos Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, Wright-Patterson AFB, OH

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/EN/TR96-03

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Generic expert systems are reasoning systems that can be used in many application domains, thus requiring application domain independence. The user interface for a generic expert system must contain an intelligence in order to maintain this domain independence and manage the complex interactions between the user and generic expert system. This paper explores the uncertainty-based reasoning contained in an intelligent user interface called GESIA. GESIA's interface architecture and dynamically constructed Bayesian network are examined in detail to show how uncertainty-based reasoning enhances the capabilities of this generic expert system user interface.

14. SUBJECT TERMS
intelligent user interface, generic expert system, uncertainty-based reasoning, bayesian network.

15. NUMBER OF PAGES
13

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

UNCLASSIFIED

AFIT/ENG/TR96-03

Air Force Institute of Technology

The PESKI Intelligent User Interface

Robert A. Harrington, Sheila Banks, and Eugene Santos Jr.

August 11, 1996

The PESKI Intelligent User Interface

Robert A. Harrington, Sheila Banks, and Eugene Santos Jr.
Department of Electrical and Computer Engineering
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433-7765
rharring@afit.af.mil

August 11, 1996

KEYWORDS: Bayesian Network, Generic Expert System, Intelligent User Interface, Uncertainty-based Reasoning

Abstract

Generic expert systems are reasoning systems that can be used in many application domains, thus requiring application domain independence. The user interface for a generic expert system must contain an intelligence in order to maintain this domain independence and manage the complex interactions between the user and generic expert system. This paper explores the uncertainty-based reasoning contained in an intelligent user interface called GESIA. GESIA's interface architecture and dynamically constructed Bayesian network are examined in detail to show how uncertainty- based reasoning enhances the capabilities of this generic expert system user interface.

1 Introduction

As generic expert systems begin to make their way into mainstream applications, research must be conducted to handle the generic system's special challenges. The greatest of all the challenges for these generic systems is maintaining application domain independence (keeping the system generic) and ensuring system performance is not linked to a specific application domain. This challenge

centers on the system's interaction with the application domain, namely the system's user, so becomes the primary focus of the system's user interface.

The user interface for a generic expert system must offer more than just a pleasing, easy to use work environment. The user interface must manage the multitude of tasks required to maintain the system's application domain independence as well as facilitate communication between the user and the system [9] [8]. These tasks include recognizing the system's application domain and user, suggesting user implemented adaptations to the interface, and adapting the interface adaptively to meet the specific domain and user's needs [7].

In order for a user interface to perform these various management tasks, it must contain an aspect of intelligence, or reasoning capability [2], that enables it to act as an intelligent assistant to the user. This is the idea behind the intelligent user interface GESIA (Generic Expert System Intelligent Assistant) for the generic expert system PESKI (Probabilities, Expert Systems, Knowledge, and Inference) [1] at the Air Force Institute of Technology. This paper discusses the basic interface architecture and dynamically constructed Bayesian network for GESIA, focusing on the use of uncertainty-based reasoning to maintain the application domain independence of the expert system.

2 Interface Reasoning Needs

In order for a user interface to perform as an intelligent assistant it must have the ability to reason. This reasoning capability is enabled by the following: collecting metrics, transitioning metrics into information, storing information, and inferencing over the stored information. These actions work hand in hand to provide an environment from which the user interface can make intelligent decisions.

The first step in creating this reasoning environment within the interface is to collect metrics based on the operations being performed on the expert system. These metrics are called interface domain metrics. Interface domain metrics can be just about any type of data that a user interface can collect from the application domain or the user. These include keystrokes, procedures used to perform tasks, user preferences, and tasks most often performed. The number and type of interface domain metrics collected is solely based on knowledge required for user interface reasoning. Information about the application domain can be acquired from a single interface domain metric or combinations of different types of metrics.

The collected interface domain metrics then needs to be transformed into some meaningful information. The information format must be based on one that the user interface requires for making decisions at a later time. This step suggests an intermediate reasoning step that develops a meaning for the metric collected. This intermediate step is contained in a knowledge based transformation algorithm that is used by the interface to convert the metric into

information.

Once the interface domain metric has been transformed, the information must be stored. The storage medium, usually a knowledge base, facilitates the reasoning process when the interface requires knowledge. An uncertainty-based scheme is a good choice for this task since it allows for efficient and affective reasoning.

When the user interface needs to make a decision, the interface will need to draw upon the knowledge stored in the knowledge base. The architectural scheme of the knowledge base will determine how intelligent and dynamic the decisions are as well as how efficient the reasoning is in terms of processing resources.

3 The Intelligent User Interface GESIA

GESIA has a layered architecture that contains three main layers: the graphical layer, the system layer, and the intelligent assistant layer (see Figure 1). The graphical layer provides the graphical interface environment, or cosmetics, for the interface, while the system layer provides a coupling between the expert system's tools and the user interface. The intelligent assistant layer is the main focus of this research.

The intelligent assistant layer has a layered architecture as well. Its layers include an adaptation layer, an adaptive layer, and communications layer. The adaptation layer recognizes particular adaptations that can be made to customize the interface to specific application domains and users [5]. The recognized adaptations are then offered to the user, and the user interface provides help in making the adaptations if the user so desires. On the other hand, the adaptive layer actually makes changes to the user interface without interaction or decision from the user [12]. The adaptive layer makes these changes based on perceived user behavior in a manner that will be explained later in this paper. The communications layer provides the methods to collect and translate interactions between the user and the expert system. Together, these three layers use reasoning to control adaptations, maintain application domain independence of the expert system, and assist the user with utilizing the expert system's functionality.

4 Intelligent Interface Reasoning

The basic model for representing GESIA's knowledge is a Bayesian network [6] [10] [3] [4]. User behavior is not deterministic, so representing user behavior in an uncertainty-based architecture is appropriate. This representation has the ability to portray a large amount of information based on the collection of only a small number of interface domain metrics, making this representation

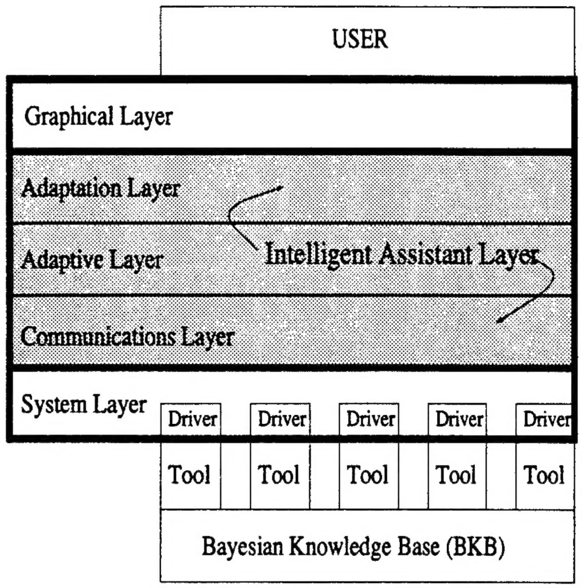

Figure 1: The GESIA architecture

important for interface reasoning efficiency. There are three types of nodes in GESIA's Bayesian network, interface learning nodes(ILNs), interface information nodes(IINs), and uncertainty support nodes(USNs). This section describes each node in the Bayesian network, and a later section will give a concrete example of how the network is used.

Interface learning nodes are used by GESIA to integrate and store the meaning of collected interface domain metrics into the Bayesian network. These nodes not only hold a specific semantic meaning but also have a set of probabilistic values attached to the meaning. The actual structure of these nodes consists of a node-unique update algorithm and a table of probabilities. The node-unique update algorithm at each node is specially designed to alter node probabilities based on the type and use of the metric received. The table of probabilities holds an entry for every user of the system plus an entry for each of the four basic user types: application user, application expert, knowledge engineer, and computer scientist. The specific semantics of each interface learning node combined with its probabilistic values allows the interface learning node to represent degrees of uncertainty in the learned information.

The interface information nodes represent the many states of the world within GESIA. These nodes feed off interface learning nodes, uncertainty support nodes, and other interface information nodes to determine probabilistic

values for the states they represent. While the interface learning nodes are the primary gateway for which learning enters the network, interface information nodes represent the application of the networks learned knowledge. User interface elicitation of knowledge targets the states represented by the interface information nodes, allowing the user interface to make intelligent decisions concerning potential adaptations.

Finally, the uncertainty support nodes store information concerning the uncertainty that the user interface will make a correct decision about a particular interface information node (system state). The structure of these nodes is much like the structure of the interface learning nodes, and there exists exactly one uncertainty support node for each and every interface information node. The probabilities stored in each of these nodes represents all the instances when the interface is wrong about inferencing over the interface information node it supports. This uncertainty is applied to its parent interface information node to alter its parents probability when its parent is targeted for knowledge elicitation by the user interface. In this way, the user interface decisions of the future will be affected by its incorrect inferences of the past.

5 GESIA Metrics and Nodes

GESIA employs the minimum set of interface domain metrics necessary to assist the user with employing expert system functions while maintaining the application domain independence of the expert system. Figure 2 displays a chart of the interface domain metrics, interface learning nodes, uncertainty support nodes, and interface information nodes GESIA supports. All of GESIA's interface domain metrics, interface learning nodes, uncertainty support nodes, and interface information nodes fall within one of three general classifications based on the information collected: functional execution, communication modes, and output styles. Together, these classifications of information provide GESIA with a considerable amount of knowledge about the outside world.

The metrics and nodes used to support the functional execution classification are used to collect, learn, and use information to answer the question "What functionality of the expert system will this user most likely use?" Every time one of the main expert system functions (knowledge acquisition, knowledge extraction, and knowledge base viewing) is executed, an interface domain metric is instantiated which represents that execution. This metric is sent to the appropriate functional execution classification learning node and is applied to each learning node's probability that is associated with the current user. Using each node's individual update algorithm, a new probability is produced and stored in the node's user table for the current user. Later, when the interface needs to question a functional execution interface information node, the probabilities of the subordinate interface learning nodes and interface information nodes are applied. The resulting probability is used by the user interface to answer the

Interface Domain Metrics	Interface Learning Nodes	Interface Information Nodes
Functional Execution	***Functional Execution***	***Functional Execution***
Knowledge Aquisition Used	User's Class Prefers Knowledge Aquisition	Using Knowledge Aquisition
Knowledge Extraction Used	User's Class Prefers Knowledge Extraction	Using Knowledge Extraction
Knowledge Viewing Used	User's Class Prefers Knowledge Viewing	Using Knowledge Viewing
	User Prefers Knowledge Aquisition	
	User Prefers Knowledge Extraction	
	User Prefers Knowledge Viewing	
Communication Modes	***Communication Modes***	***Communication Modes***
Structured Text Used	User's Class Prefers Structured Text	Using Structured Text
Natural Language Used	User's Class Prefers Natural Language	Using Natural Language
Graphical Manipulation Used	User's Class Prefers Graphical Manipulation	Using Graphical Manipulation
	User Prefers Structured Text	
	User Prefers Natural Language	
	User Prefers Graphical Manipulation	
Output Styles	***Output Styles***	***Output Styles***
Best Response Requested	User's Class Prefers Best Response	Receiving Best Response
Best 5 Responses Requested	User's Class Prefers Best 5 Responses	Receiving 5 Best Responses
Best 10 Responses Requested	User's Class Prefers Best 10 Responses	Receiving 10 Best Responses
	User Prefers Best Response	
	User Prefers Best 5 Responses	
	User Prefers Best 10 Responses	

Figure 2: A table of GESIA metrics and nodes

question.

The information collected for the functional execution classification are used by the interface for two purposes. First, the information allows the interface to perform interface initiated abstraction of seldom used functions. Second, the functional execution information supports decisions for the communication modes classification.

The metrics and nodes used to support the communication modes classification are used to collect, learn, and use information to answer the question "What type of communication does this user prefer when utilizing the powers of the expert system?" The interface domain metric is collected each time a communication mode (natural language, structured text, or graphical manipulation) of the interface is used to perform an expert system task. This metric must be collected each time a communication mode is activated to translate between the user and the expert system and is processed much like the functional execution metrics. An example of the use for this information is if the user has a high probability of using a particular communication mode and the user starts an expert system function, the interface can automatically bring up that communication mode. If the probabilities are close between two modes, the system initiates a query to the user asking which mode the user wants. In this way, the user is assisted by the user interface in choosing a communication mode for a given expert system function.

The metrics and nodes used to support the output styles classification are used to collect, learn, and use information to answer the question "How many of the best matches from a query will this user prefer?" The interface domain metric is collected by obtaining how many outputs the user requests, or how many outputs the user accepts if a reasoned number of outputs is returned by the interface. Again, this metric is processed much like the function execution metrics. The output styles information is especially useful when the user fails to specify what style of output is required for a specific execution of an expert system query. As with the communication modes, if the user interface finds probabilistic tendencies toward a particular output style, the user interface will automatically return the most probable desired output style. If the probabilities are close, the user interface will query the user for clarification.

The interface domain metrics, interface learning nodes, uncertainty support nodes, and interface information nodes are combined to dynamically construct a Bayesian network (see Figure 3). This network represents the knowledge that the user interface collects dynamically, as the user utilizes the expert system.

As previously mentioned, the interface learning nodes (shown in regular ovals, Figure 3) occupy the fringe of the structure and offer a means to input newly learned information into the network. These interface learning nodes lend dependencies to corresponding interface information nodes (shown in bold ovals, Figure 3), creating new probabilities for the interface information nodes. These new probabilities are supplemented by uncertainty stored in the uncertainty support nodes (shown in squares labeled USN, Figure 3). These USNs abstractly

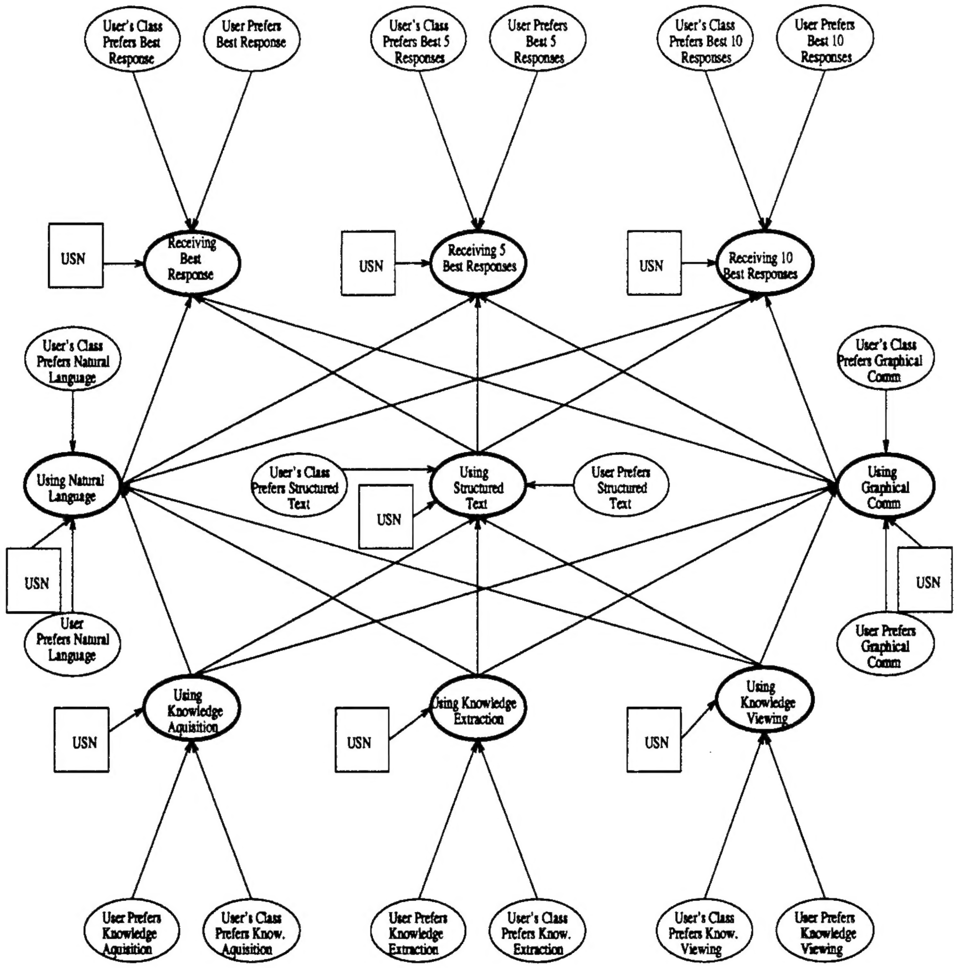

Figure 3: The GESIA network

measure how a domain can determine the interface adaptations required. The USN measurements are applied to the interface information nodes to alter the final probabilities of the supported states.

These dependencies can also be passed to other interface information nodes. In Figure 3, functional execution interface information nodes, interface learning nodes, and uncertainty support nodes all feed dependencies into the communication mode interface information nodes. Likewise, communication mode interface information nodes, interface learning nodes, and uncertainty support nodes all feed dependencies into the output style interface information nodes. Together, these relations add probabilities as they trace through the network to influence the final probability of the interface information node being questioned by the interface.

6 Example of Interface Information Node Query

This example of a simple network demonstrates how the network learns and how the learned data can be used to create a probability for a possible state. Figure 4 depicts the network used in this example. Notice there is only one IIN, named "User is Using Graphical Communication" (UGC). There is also the supporting USN, named "Uncertainty User is Using Graphical Communication"(UUGC). Finally, there are two ILNs, named "User's Class Prefers Graphical Communication"(CPGC) and "User Prefers Graphical Communication"(UPGC). For this example, let's say a user, login TOM, has logged onto PESKI through GESIA. GESIA's network recovers all the learned data about TOM from storage and sends the data to the appropriate ILNs and USNs in the network.

With the network loaded, TOM begins to use GESIA. As TOM performs actions through the interface, the interface records TOM's behavior by updating network ILNs and USNs. For example, in Figure 4, if TOM chooses to use graphical communication from the communication mode menu of the interface, the interface will update CPGC and UPGC. Thus, TOM's behavior is captured.

Later, if the interface wants to guess what communication mode TOM will choose, the interface will query the UGC for the node's probability. This probability is calculated by combining the probabilities of CPGC, UPGC, and UUGC. The probabilities are combined using the following method. First, a truth table is constructed that lists all the possible combination of the truthfulness of CPGC and UPGC. Therefore,

P(UGC=T — CPGC=T, UPGC=T):=1.00,
P(UGC=T — CPGC=T, UPGC=F):=0.65,
P(UGC=T — CPGC=F, UPGC=T):=0.65, and
P(UGC=T — CPGC=F, UPGC=F):=0.00.

Notice if the probabilities that CPGC and UPGC are both true then the probability of UGC being true is 1.00, and if the probabilities that CPGC and

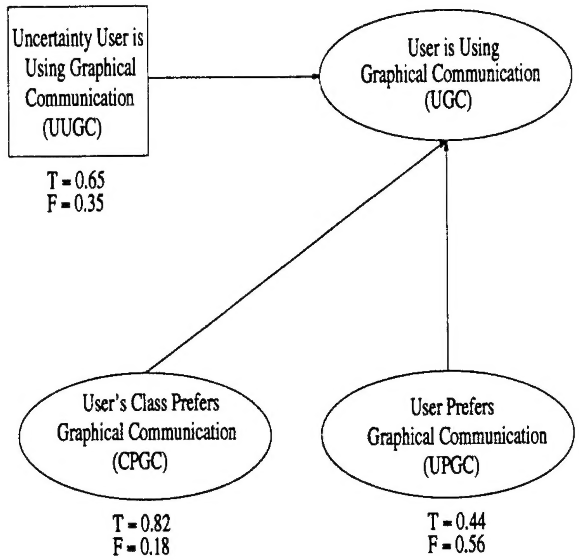

Figure 4: A partial network for query example

UPGC are both false then the probability of UGC being true is 0.00. In cases when the probability of an IIN is not absolute, the uncertainty of the truthfulness of the IIN must be supported by it's USN. Once the truth table is constructed, the probabilities may be combined using Bayes theorem [6] [10] [3] [4]:

$$
\begin{aligned}
P(UGC=T) = {} & P(UGC,CPGC,UPGC) \\
& + P(UGC,not(CPGC),UPGC) \\
& + P(UGC,CPGC,not(UPGC)) \\
& + P(UGC,not(CPGC),not(UPGC)). \\
& = (1.00*0.82*0.44) + (0.65*0.18*0.44) \\
& + (0.65*0.82*0.56) + (0.00*0.18*0.56).
\end{aligned}
$$

Therefore, UGC-T=0.7108 or 71 percent. Given this result, the user interface has acquired a mathematically sound method for which to capture user behavior and then convert it into a representation from which the user interface may reason about future user intent.

7 Design Problems

While this reasoning architecture enhances the abilities of the user interface to intelligently adapt to the application domain and the user, there are some shortfalls that must be overcome. These shortfalls revolve around metric collection, information storage, and metric update control.

Metric collection must be accomplished as sparingly as possible since a metric must be processed every time it is collected. The collection of too many metrics too often overburdens the tasks the user is trying to perform. As the speed of processing diminishes, so too does the value of the user interface's intelligent adaptations.

Storage of all the user's probabilities at each of the information nodes is relatively trivial when only few users employ the expert system. However, as the expert system gains users or changes users, more records will have to be stored at each information node. This storage must be managed to avoid overtaking storage memory resources.

A control algorithm must be in place to handle the possibly constant flow of metrics that are collected and applied to the interface learning nodes. This problem is not unlike the process control problem operating systems designers face [11]. The control algorithm must ensure the interface information nodes are accessed and updated in a deterministic fashion to ensure accuracy of the interface's reasoning.

8 Results and Conclusions

Constructing a reasoning architecture into a user interface in order to facilitate intelligent interface decision making can be performed using Bayesian networks. The uncertainty-based principles of Bayesian networks aid in representing the uncertainty a user interface encounters when assessing what the user needs. Emphasis on generic expert system user interface design, and the reasoning architecture behind it, is important to keep generic expert systems generic.

The current implementation of GESIA uses only a few specific metrics to represent simple information about the application domain and the system's users. This basic interface architecture will be enhanced to represent and reason with more complex structures and ideas. Reasoning will be tied closer to the natural language interpreter of the user interface, creating a dialogue capability between the user and the user interface, allowing the user to communicate with the user interface using more abstract communication.

References

[1] Darwyn O. Banks. Acquiring consistent knowledge for bayesian forests. Master's thesis, Graduate School of Engineering, Air Force Institute of

Technology, Wright-Patterson AFB OH, 1995.

[2] Edwin Bos, Carla Huls, and Wim Claassen. Edward: full integration of language and action in a multimodal user interface. *International Journal of Human-Computer Studies*, 40:473–495, 1994.

[3] Eugene Santos Jr. Computing with bayesian multi-networks. Technical Report AFIT/EN/TR93-10, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, 1993.

[4] Eugene Santos Jr and Eugene S. Santos. Representing and reasoning with bayesian knowledge-bases. *submitted to Journal of the ACM*, 1996.

[5] Eliezer Kantorowitz and Oded Sudarsky. The adaptable user interface. *Communications of the ACM*, 11(32), November 1989.

[6] Kathryn Blackmond Laskey. The bounded bayesian. In *Proceedings of the Eight Conference on Uncertainty in Artificial Intelligence*, 1992.

[7] Beth Meyer. Retail user assistant: Evaluation of a user-adapted performance support system. In *Lecture Notes in Computer Science, 4th International Conference, EWHCI'94*, 1994.

[8] Irene Neilson and John Lee. Conversations with graphics: implications for the design of natural language/graphics interfaces. *International Journal of Human-Computer Studies*, 40:509–541, 1994.

[9] Reinhard Oppermann. Adaptively supported adaptivity. *International Journal of Human-Computer Studies*, 40:455–472, 1994.

[10] Judea Pearl. *Probabalistic Reasoning in Inteligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, CA, 1988.

[11] Mukesh Singhal and Niranjan G. Shivaratri. *Advanced Concepts in Operating Systems.* McGraw-Hill, Inc, New York, NY, 1994.

[12] James E. Trumbly. Productivity gains via an adaptive user interface: an empirical analysis. *International Journal of Human-Computer Studies*, 40:63–81, 1994.